REMARKS

Claims 1-21 remain in the application. Claims 1-5 are hereby amended. No new matter is being added.

Claim Rejections -- 35 U.S.C. § 103

Claims 1-21 were rejected as being unpatentable over Vosbury (USP 5,138,708) in view of Vrba et al (USP 5,845,060). Applicants respectfully traverse this rejection.

A) Both Vosbury and Vrba et al relate to the conventional technique where two or more CPUs execute the same instruction stream

Applicants respectfully submit that both Vosbury and Vrba et al relates to a technique where fault tolerance is provided for a conventional program by **running the same program on two or more CPUs**.

This conventional technique is discussed under the Description of the Background Art of the present application. As discussed therein,

> Previous solutions for providing fault tolerance in digital processing are either hardware based, software based, or some combination of both. **Fault tolerance may be provided in hardware by running two full central processing units (CPUs) in lockstep, or three CPUs in a "voting" configuration.** For example, a system may employ three CPUs executing the same instruction stream, along with three separate main memory units and separate I/O devices which duplicate functions, so if one of each type of element fails, the system continues to operate. **Unfortunately, such systems include tremendous system overhead, not only in terms of the number of CPUs required, but also in terms of the infrastructure supporting the CPUs** (memory, power, cooling systems, and so on).
>
> (Emphasis added.)

Note that such a conventional solution has the disadvantage of needing two or more CPUs and supporting infrastructure.

In particular, Vosbury describes the architecture of its system as follows on column 2, lines 24-33.

> Referring to FIG. 1, the architecture of a highly fault-tolerant processor (HFTP) 10 implemented in accordance with the present invention is illustrated. **The HFTP is preferably implemented on one chip and includes a central processing unit (CPU) 11 and a CPU 12. The CPUs 11 and 12 are identical with respect to each other in hardware and software and in operation execute the same instruction stream** in lock step. The CPUs 11 and 12 are tightly coupled with respect to each other.
>
> (Emphasis added.)

As shown above, Vosbury teaches two CPUs "executing the same instruction stream" to provide fault tolerance.

Similarly, Vrba et al describes its computer system as follows in the Abstract.

> A fault-tolerant computer system employing **multiple CPUs executing the same instruction stream** under independent clock cycle timing. The CPUs deterministically execute the instructions internally until input or output operations require access to memory or devices which are not synchronous with the local CPU clock.
>
> (Emphasis added.)

As shown above, Vrba et al teaches "multiple CPUs executing the same instruction stream" to provide fault tolerance.

B) Claimed invention operates on a different principle from Vosbury and Vrba et al. and enables fault tolerance to be targeted to certain code (with untargeted code being executed without fault tolerance)

The principle of the claimed invention is discussed in the specification. For example, the specification recites as follows on page 3, line 25 through page 4, line 9.

> An embodiment of the present invention utilizes **special versions of certain CPU instructions** to provide fault tolerance in a targeted manner. Specific operations within an application may be **targeted for fault tolerance**, while other operations (or other entire programs) may be performed without the overhead due to redundancy checking.
>
> Such targeted fault tolerance has various advantages over prior solutions. **It may be selectively applied to system processes, instead of being applied to all system processes. There are some processes that are not critical enough to warrant the dedication of such resources, or that are desirable to run as fast as possible (without being slowed down by redundancy checking). For example, a print spooler program is unlikely to be critical enough to need fault tolerance.** In accordance with an embodiment of the invention, targeted fault tolerance allows such a non-critical program to be written without the special redundancy-checking instructions, so that the non-critical program does not unnecessarily tie up valuable system resources. On the other hand, critical programs or processes requiring redundancy checking may be written using the special instructions so as to provide fault-tolerant execution thereof. The choice may be left up to the application programmer.
>
> (Emphasis added.)

As described above, the claimed invention relates to **targeting** fault tolerance to certain code or processes. This is done by providing **a same instruction in two different versions, with one version being a special "fault-tolerant" instruction and another version being an ordinary instruction which is performed without redundancy checking.**

C) Claimed invention does not require multiple CPUs and, instead, is performed within a single CPU

As discussed above, both Vosbury and Vrba et al provide a fault tolerant solution by using **multiple CPUs** executing the same instruction stream to provide fault tolerance. In contrast, the claimed invention operates on a different principle and is performed **within a single CPU**.

The Summary in the specification states as follows on page 2, lines 15-20.

> Another embodiment of the invention pertains to **a method for targeted fault-tolerant computing in a central processing unit (CPU)**. The method includes decoding a fault-tolerant version of an instruction to generate a first op code and decoding a non-fault-tolerant version of the instruction to generate a second op code. The first op code is executed with redundancy checking. The second op code is executed without redundancy checking.
>
> (Emphasis added.)

As described above, the invented technique of targeted fault-tolerant computing is performed within a single CPU.

D) Under MPEP 2143.01, THE PROPOSED MODIFICATION CANNOT CHANGE THE PRINCIPLE OF OPERATION OF A REFERENCE

MPEP 2143.01 states as follows.

THE PROPOSED MODIFICATION CANNOT CHANGE THE PRINCIPLE OF OPERATION OF A REFERENCE

If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *In re Ratti*, 270 F.2d 810, 123

USPQ 349 (CCPA 1959) (Claims were directed to an oil seal comprising a bore engaging portion with outwardly biased resilient spring fingers inserted in a resilient sealing member. The primary reference relied upon in a rejection based on a combination of references disclosed an oil seal wherein the bore engaging portion was reinforced by a cylindrical sheet metal casing. Patentee taught the device required rigidity for operation, whereas the claimed invention required resiliency. The court reversed the rejection holding the "suggested combination of references would require a substantial reconstruction and redesign of the elements shown in [the primary reference] as well as a change in the basic principle under which the [primary reference] construction was designed to operate." 270 F.2d at 813, 123 USPQ at 352.).

E) Discussion of specific claims

Amended claim 1 now recites as follows.

1. **A central processing unit (CPU)** for **targeted** fault-tolerant computing, the CPU comprising:

   decode circuitry configured to decode a fault-tolerant version of an instruction and a non-fault-tolerant version of the instruction distinctly from each other; and

   execution circuitry configured to execute the fault-tolerant version of the instruction with redundancy checking and to execute the non-fault-tolerant version of the instruction without redundancy checking.

(Emphasis added.)

As shown above, claim 1 requires "decode circuitry configured to decode a fault-tolerant version of an instruction and a non-fault-tolerant version of the instruction." In addition, claim 1 requires "execution circuitry configured to execute the fault-tolerant version of the instruction with redundancy checking and to execute the non-fault-tolerant version of the instruction without redundancy

checking." Both the decode circuitry and the execution circuitry are in the single claimed CPU.

Applicants respectfully submit that decode circuitry configured to **decode two versions of a same instruction** is not disclosed or taught in either Vosbury or Vrba et al. Furthermore, applicants respectfully submit that execution circuitry configured to **execute two versions of a same instruction** is not disclosed or taught in either Vosbury or Vrba et al. Moreover, applicants note that the claim requires the decode circuitry and the execution circuitry to be within the **single claimed CPU**.

As discussed above, both Vosbury and Vrba et al operate on a **different principle** than the claimed invention. Instead of a **single CPU** decoding and executing **multiple versions of the same instruction** to target fault tolerance, both Vosbury and Vrba et al teach using **multiple CPUs** to provide fault-tolerant execution of an entire instruction stream. Therefore, under MPEP 2143.01, applicants respectfully submit that claim 1 cannot be rendered *prima facie* obvious by the teachings of Vosbury and Vrba et al.

For at least the above-discussed reasons, applicants respectfully submit that claim 1 is now patentably distinguished over the cited art.

Claims 2-5 depend from claim 1. Hence, applicants respectfully submit that claims 2-5 are now patentably distinguished over the cited art for at least the same reasons discussed above in relation to claim 1.

Similar to claim 1, claim 6 recites as follows.

6. A method for targeted fault-tolerant computing in a central processing unit (CPU), the method comprising:
decoding a first op code corresponding to a fault-tolerant version of an instruction;
decoding a second op code corresponding to a non-fault-tolerant version of the instruction;
executing the first op code with redundancy checking; and

executing the second op code without redundancy checking.

(Emphasis added.)

As shown above, claim 6 requires decoding two op codes for the same instruction, the first op code being a fault-tolerant version and the second op code being a non-fault-tolerant version. In addition, claim 6 requires executing the first op code with redundancy checking and executing the second op code without redundancy checking. Both the decoding and executing are performed in a single CPU.

As discussed above, both Vosbury and Vrba et al operate on a **different principle** than the claimed invention. Instead of a **single CPU** decoding and executing **multiple versions of the same instruction** to target fault tolerance, both Vosbury and Vrba et al teach using **multiple CPUs** to provide fault-tolerant execution of an entire instruction stream. Therefore, under MPEP 2143.01, applicants respectfully submit that claim 6 cannot be rendered *prima facie* obvious by the teachings of Vosbury and Vrba et al.

Hence, applicants respectfully submit that claim 6 is patentably distinguished over the cited art.

Claims 7-13 and 21 depend from claim 6. Hence, applicants respectfully submit that claims 7-13 and 21 are patentably distinguished over the cited art for at least the same reasons discussed above in relation to claim 6.

Similarly, claim 14 recites as follows.

14. A computing apparatus for targeted fault-tolerant computing, the apparatus comprising:
means for **decoding a first op code corresponding to a fault-tolerant version of an instruction and a second op code**

corresponding to a non-fault-tolerant version of the instruction;
redundant means for executing the first op code; and
non-redundant means for executing the second op code.

(Emphasis added.)

As shown above, claim 14 requires means for decoding two op codes for the same instruction, the first op code being a fault-tolerant version and the second op code being a non-fault-tolerant version. In addition, claim 14 requires redundant means for executing the first op code and non-redundant means for executing the second op code.

As discussed above, both Vosbury and Vrba et al operate on a **different principle** than the claimed invention. Instead of means for decoding and executing **multiple versions of the same instruction** to target fault tolerance, both Vosbury and Vrba et al teach using **multiple CPUs** to provide fault-tolerant execution of an entire instruction stream. Therefore, under MPEP 2143.01, applicants respectfully submit that claim 14 cannot be rendered *prima facie* obvious by the teachings of Vosbury and Vrba et al.

Hence, applicants respectfully submit that claim 14 is patentably distinguished over the cited art.

Claim 15 depends from claim 14. Hence, applicants respectfully submit that claim 15 is patentably distinguished over the cited art for at least the same reasons discussed above in relation to claim 14.

Similarly, claim 16 recites as follows.

16. A computer program product comprising a computer-usable medium having computer-readable code embodied therein, the computer program product including:

a first type of computer-readable instructions to be executed with redundancy checking; and

a second type of computer-readable instructions to be executed non-redundantly.

(Emphasis added.)

As shown above, claim 16 requires a program product to have two types of instructions, the first type to be executed with redundancy checking, and the second type to be executed non-redundantly.

As discussed above, both Vosbury and Vrba et al operate on a **different principle** than the claimed invention. Instead of **two types of instructions** (one to be executed with redundancy checking and the other to be executed non-redundantly) to target fault tolerance, both Vosbury and Vrba et al teach using **multiple CPUs** to provide fault-tolerant execution of an entire instruction stream. Therefore, under MPEP 2143.01, applicants respectfully submit that claim 16 cannot be rendered *prima facie* obvious by the teachings of Vosbury and Vrba et al.

Hence, applicants respectfully submit that claim 16 is patentably distinguished over the cited art.

Claims 17-20 depend from claim 16. Hence, applicants respectfully submit that claims 17-20 are patentably distinguished over the cited art for at least the same reasons discussed above in relation to claim 16.

Conclusion

For at least the above reasons, it is believed that the pending claims are now patentably distinguished over the cited art. The Examiner is invited to telephone the undersigned at (408) 436-2111 for any questions.

If for any reason an insufficient fee has been paid, the Commissioner is hereby authorized to charge the insufficiency to Deposit Account No. 50-2427.

Respectfully submitted,
Ken Gary Pomaranski, et al.

Dated: July 12, 2006

James K. Okamoto, Reg. No. 40,110
Okamoto & Benedicto LLP
P.O. Box 641330
San Jose, CA 95164
Tel.: (408)436-2110
Fax.: (408)436-2114

CERTIFICATE OF MAILING			
I hereby certify that this correspondence, including the enclosures identified herein, is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.			
Signature:			
Typed or Printed Name:	James K. Okamoto	Dated:	July 12, 2006